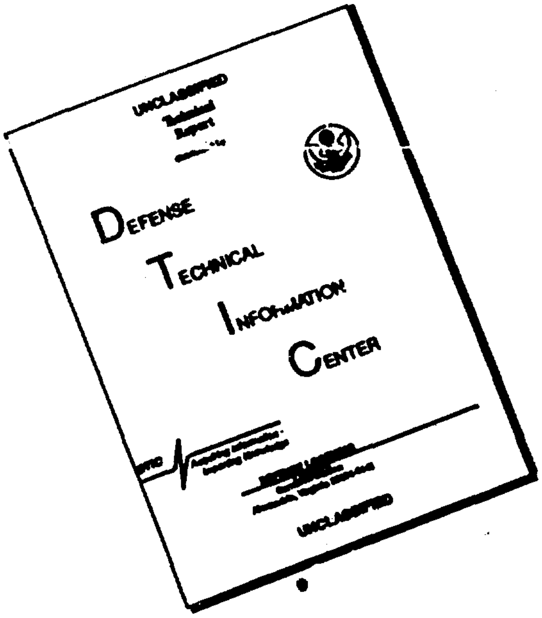
DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

AD 712134

INTERIM REPORT
1013.1 1
QUANTIFICATION OF INFORMATION STORAGE AND RETRIEVAL METHODOLOGIES

---o---

CONTRACT NUMBER N00014-70-C-0044

RESEARCH SPONSORED BY THE OFFICE OF NAVAL RESEARCH UNDER NRL PROJECTS RF018-02-41 AND RR003-09-41-502

5 JUNE 1970

ANALYTICS, INCORPORATED
179 WASHINGTON LANE
JENKINTOWN, PENNSYLVANIA 19046

QUANTIFICATION OF INFORMATION STORAGE
AND RETRIEVAL METHODOLOGIES

This Interim Report 1013.1-1 represents completion of Phase 1 of work under contract N00014-70-C-0044 with the Office of Naval Research for the U.S. Naval Research Laboratory. This report is composed of two papers which represent the areas of concentration of this study as directed by the Contract Scientific Officer and his designated representatives.

NOISE - ITS EFFECT ON DEPTH OF FILE SEARCH

by: Morris Plotkin
and
Samuel D. Epstein

This work was performed by Analytics Incorporated under Contract Number N00014-70-C-0044 from the Office of Naval Research for the Information Sciences Division of the U.S. Naval Research Laboratory, Washington, D.C.

ABSTRACT

This paper presents the results of a set of Monte Carlo computations designed to show the general behavior of the efficiency of probabilistic information retrieval systems as a function of human-variability noise. The total amount of noise, the combination of noise produced in indexing documents and in formulating requests, is the independent variable. The effect of noise is measured by the fraction of the file that must be retrieved in order to obtain the document that in the absence of noise would be retrieved first. Computations are made for an idealized system in which the index and request vectors are normalized and have uniform distributions; however, the method could accommodate other distributions. The results show how, for a fixed amount of noise, the depth of file search decreases with increasing numbers of index categories for each constant ratio of terms specified in the query to index categories in the space. Also, for a fixed number of index categories, the way in which the fraction of file searched decreases with the number of index terms in the query is shown.

INTRODUCTION

Probabilistic indexing techniques as first introduced by Maron and Kuhns [1] are capable of a wider variety of responses than Boolean systems. In a probabilistic retrieval system each document D_i is assigned an index vector V_i whose elements quantify the degree to which each index term describes the document. Likewise, request vectors describe information needs in terms of the same index space. The relevance of any document D_i to a request R is then a function of V_i and R, and the response to a request is an ordering of the documents according to their relevance to that request.

The probabilistic system allows the elements v_{ij} of the index vector V_i and the elements r_k of the request vector R to take on any value in the range (0,1). Thus a relevance measure such as $r=V_i \cdot R$ (suitably normalized) associates relevance with the intuitive concept of distance between document and request in Euclidean space. Stiles [2] and Shumway [3] further expand the range of probabilistic techniques by introducing clustering, the grouping of index terms by statistical association of the indexed documents. Jones and Needham [4] demonstrate a system based upon matching request and document groups.

In the present paper, we consider the effect of human variability noise in the generation of index and request vectors upon the efficiency of probabilistic retrieval systems.

Definition of the Problem

Let D be an arbitrary document with index vector V. Because of indexing noise, it is assigned the index vector V^n instead. A user has an information need that is exactly satisfied by the document D. He should therefore express his need by the request vector R, where ideally R=V, but because of request noise, he specifies R^n instead. The retrieval system ranks each document D_i in the file according to the value of the inner product $V_i^n \cdot R^n$ of its index vector V_i^n with the request vector R^n. In the absence of noise, the desired document D would have been ranked first, but with noise present, it may well be outranked by other documents. The purpose of the computations here reported is to investigate how the ranking of D is affected by the amount of noise as measured by the inner product

$$r = R^n \cdot V^n \qquad (1)$$

where both index vector V^n and request vector R^n are normalized. The effect of indexing noise and request noise is expressed by the departure of r from the noiseless-case value of r=1.

The Nature of the Noise

Indexing noise is the variability in assigning an index vector to a given document. To measure indexing noise experimentally, give the same document to a number of people for indexing, assume the mean to be the correct index vector for the document, and observe the departures of the individual index vectors from the mean. This procedure gives variations about the mean and ignores variations of the mean which can be made small by using a sufficiently large number of indexers. Similarly, request noise exists because two people with the same need for in-

formation do not always express the need by means of identical request vectors.

This paper reports no measurements of either indexing noise or request noise; instead, the depth of file search is presented as a function of the noise which is measured by the angular distance between V^n and R^n.

Use of a Document-Generating Distribution

In a small file it is not important that the probabilistic information retrieval system perform extremely well. If it does not, a small number of documents are examined unnecessarily. In a large file, the penalty for poor performance is greater. The file size is therefore a parameter affecting system effectiveness. To eliminate this parameter, the set of documents is represented by a generating distribution instead of a finite set. Rather than counting the number of documents that outrank the desired document D, that is, the number of V_i^n for which

$$V_i^n \cdot R^n > V^n \cdot R^n = r, \tag{2}$$

the computation will estimate the probability that equation (2) is satisfied by a D_i, with index vector V_i^n, randomly selected from the generating distribution.

Choice of the Document-Generating Distribution

The ranking of documents that the system produces in response to a request vector is unaffected if it is multiplied by a positive scalar. Therefore, there is no loss in generality in normalizing the request vectors so that their Euclidean lengths, the square-root of the sum of the squares of the vector elements, are all unity. The request vectors are assumed to be normalized in this manner.

The index vectors are also assumed to be normalized. This assumption is not innocuous; it has physical implications. It implies, for example, that every document is equally worthy of retrieval, needing only the proper request vector to make it the first-ranked document in the response. In particular, this assumption implies that a document dealing with a wide variety of subjects -- a handbook, for example -- is not accorded greater or lesser prominence in the retrieval system than a highly-specialized document. Under the assumption of normalization, the index vectors may be represented geometrically in Euclidean n-space as vectors emanating from the origin and with terminus on the positive orthant (including boundary) of the unit sphere -- n being, for the moment anyway, the number of index terms. The number n is, like the noise level, a parameter in the results presented in this paper.

It is further assumed that the index terms are uniformly distributed over the positive orthant of the unit sphere. This is a powerful assumption, but not as drastic as it first sounds, for the following reason. If n is the total number of index terms in the system, as suggested above, the assumption of an even distribution of index vectors is totally unrealistic because it is known that index terms commonly occur in clusters. But if the parameter n is interpreted in the results as the number of index terms in one of the clusters, the assumption is less objectionable since a desirable selection of index terms within a cluster is the selection giving uniform scope to each term. The results under this interpretation show the fraction of documents in the cluster that must be retrieved to reach the document that would be retrieved first in a noiseless system.

The assumptions of normalization and uniform distribution on the index vectors are made for two reasons: first, they provide simplification in the mathematics and second, they do not conflict with what is known. If there were good reason to believe in any other specific distribution of the index vectors, simplicity of mathematics could be sacrificed for the sake of realism, and computations such as those here reported could be performed using the more realistic distribution.

The Computation Problem

As a result of the assumptions set forth above, the problem of computing the efficiency of the retrieval system has a simple geometric representation in n-dimensional Euclidean space. Let S denote the n-dimensional unit sphere: the set of points $(x_1,\ldots,x_n)$ for which

$$x_1^2+\ldots+x_n^2 = 1. \qquad (3)$$

Let S+ represent the positive orthant of S, the set of points satisfying (3) with

$$x_j \geq 0, \qquad j=1,\ldots,n\ . \qquad (4)$$

The assumptions on the index vectors is that they are uniformly distributed over S+. The x's are, of course, the weights used in the index and request vectors.

Let O be the origin, the center of the unit sphere S, and let P and Q denote any points on S. Then the angle POQ is called the angular distance between P and Q.

Let $M(\Theta/Z)$ denote the measure, the (n-1)-dimensional "area", of those points that are both in S+ and within angular distance θ of Q, where Q is an arbitrary point in

S+. If M(S+) is the measure of S+, then the rat

$$\frac{M(\theta/Q)}{M(S+)} \qquad (5)$$

is the fraction of documents, in the index-term cluster represented by S, within angular distance Θ of Q; it is the fraction of the documents whose (inner product) relevance number r with respect to a request vector Q is at least

$$r = \cos\Theta \ .$$

If Q is allowed to range with uniform distribution over S+, the mean value of the ratio (5) is the average fraction of documents that would have to be retrieved to insure finding all documents displaced by angular distance Θ from the position of the request vector as a result of noise effects.

But uniform distribution of Q over S+ is unnecessarily restrictive, therefore, there is introduced another parameter k, the number of non-zero index-term weights in the request vector Q, where $k \leq n$. For example, if n=8 and k=4, the request vectors are of the form $(x_1, \ldots, x_n)$ where

$$x_a^2+x_b^2+x_c^2+x_d^2 = 1, \qquad a \neq b \neq c \neq d \qquad (6)$$

and where a, b, c, and d are any four different selections from the eight numbers (1,...,8), the other four weights being zero. Since in the results it does not matter which k of the n weights are non-zero, it will be assumed henceforth that $x_1, \ldots, x_k$ are the non-zero weights and $x_{k+1}, \ldots, x_n$ are zero. In the computations here reported, the request vectors Q are assumed to be uniformly distributed over the positive orthant of the k sphere

$$x_1^2+\ldots+x_k^2 = 1. \qquad (7)$$

Let T denote the distribution of Q-vectors just described, and let $\overline{M}(\Theta/T)$ denote the mean of $M(\Theta/Q)$ over T. Then the ratio

$$\frac{\overline{M}(\Theta/T)}{M(S+)} \qquad (8)$$

is the average fraction of documents that would have to be retrieved, under the assumptions of the computation, to reach a document displaced by noise effects through an angular distance Θ from an initial position coincident with the request vector. Of course, if a more realistic distribution T were known for the request vectors, it could be used in the ratio (8) in place of the uniform distribution presently used.

The Appendix presents the details of the computational procedure used. The program to perform this computation has not been included, but is available.

Results

The results of the Monte Carlo computations are presented as the curves of Figures 1 through 4. These curves plot the fraction F(r) of the subfile that must be searched against the relevance number

$$r=\cos\Theta \qquad (9)$$

where

$$F(r) = \frac{\overline{M}(\Theta/T)}{M(S+)} \qquad (10)$$

as in (8).

To summarize the meaning of the curves: if there is a document D that exactly matches the user's information needs but, because of indexing noise and request noise, the actual index vector for D and the actual request vector are at angular distance θ apart, F(r) is the fraction of the subfile that must be searched, on the average, before the user finds D. "Subfile" here means the portion of the file that deals with the cluster of index-terms into which D falls. The parameters n and k are, respectively, the number of index-terms in the cluster and the number of those index-terms that occur with non-zero weight in the request vector, $k \leq n$.

The Monte Carlo sample sizes used were inadequate for reliable determination of F(r) for large θ; the portions of the curves shown dotted are extrapolations.

The curves clearly show how, for fixed values of the ratio k/n, the fraction of the file that need be searched <u>for a given value of θ</u> decreases with increasing values of n, the number of index-terms in the cluster. However, it is to be expected that the human indexing noise represented by θ increases with n, and does so possibly fast enough to outweigh the decrease shown for a fixed θ. Comparison among the four figures shows how, for fixed n, the fraction of the file that must be searched decreases with k, the number of index-terms used in the query; that is, with the degree of specialization of the document within the cluster.

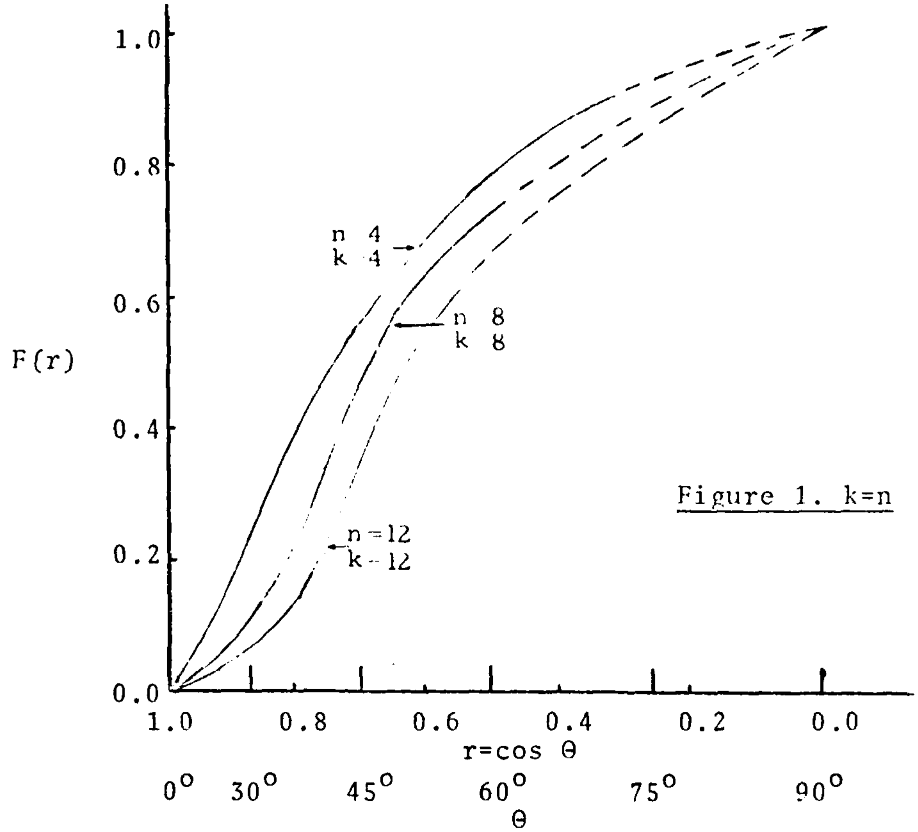


Figure 1. k=n

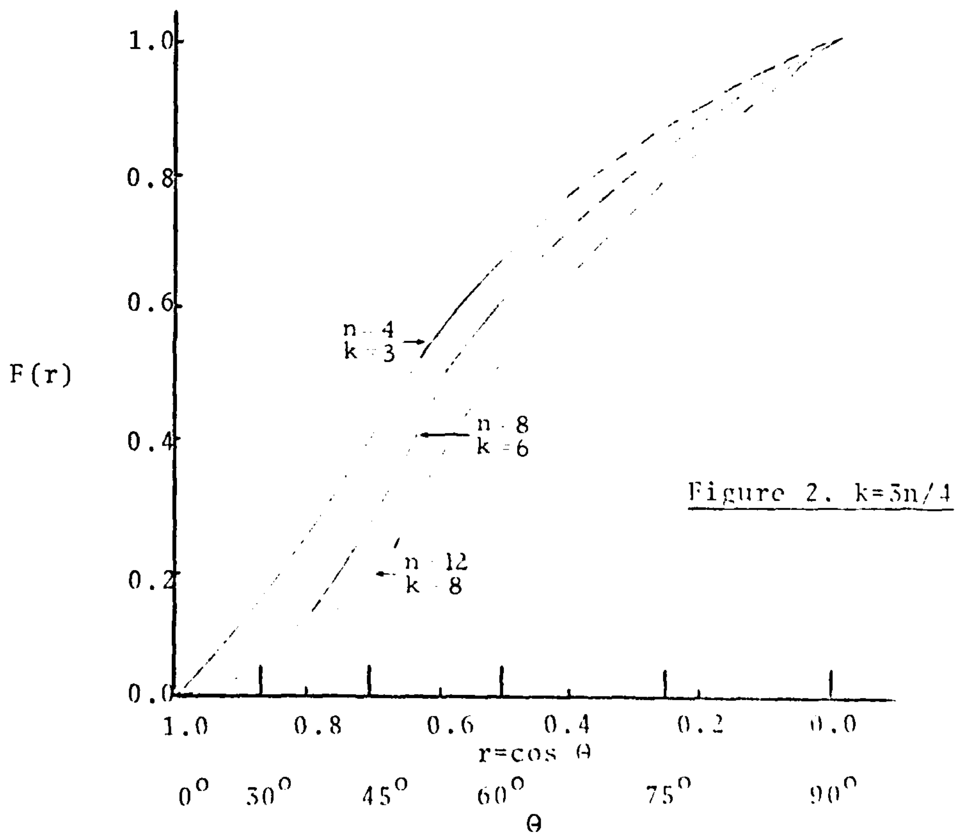


Figure 2. k=3n/4

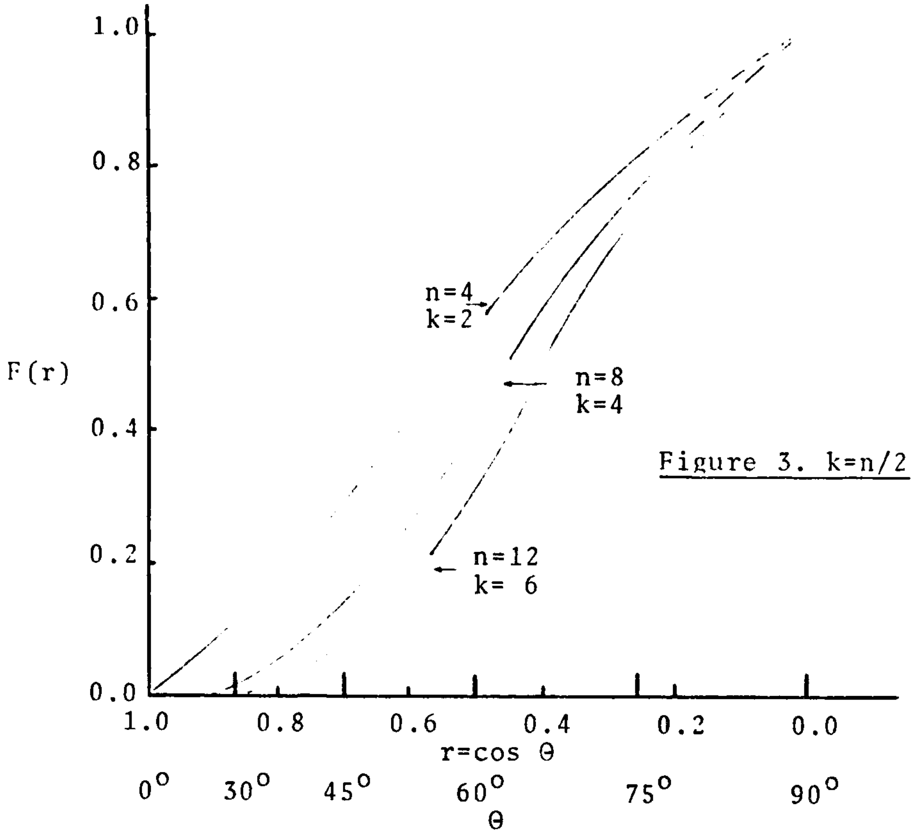


Figure 3. k=n/2

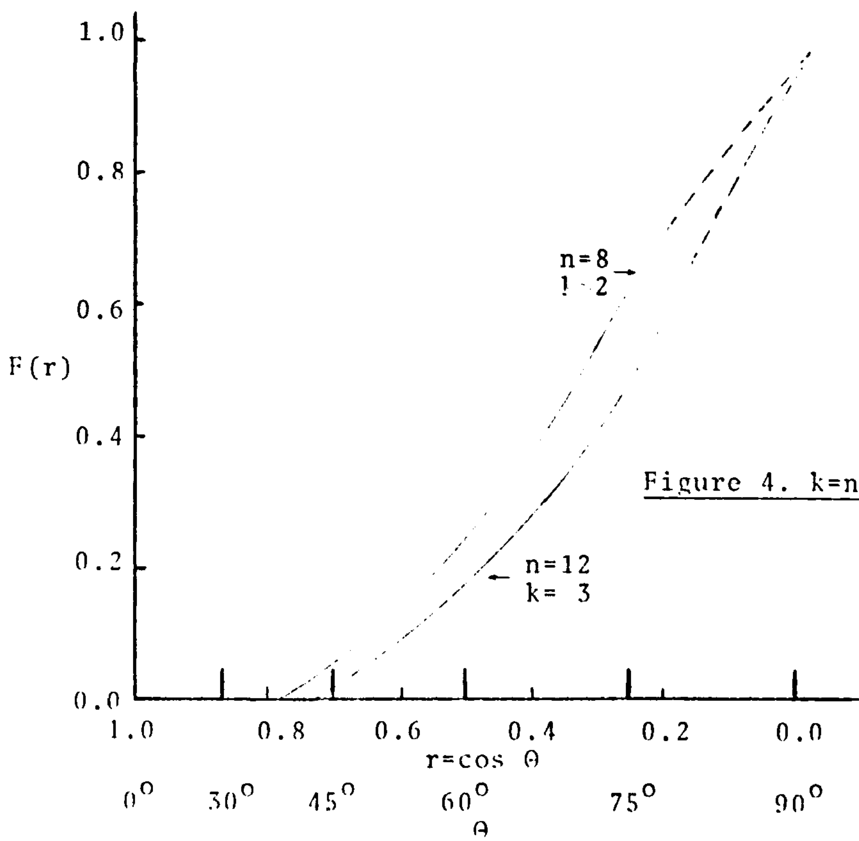


Figure 4. k=n/4

Summary

The experiment considered in this paper provides qualitative characterization for the efficiency of probabilistic information retrieval systems. Although an idealized system has been employed, the methodology presented can be extended to actual systems and made highly dependent on the particular properties of a real data base. With the advent of non-Boolean retrieval methodologies over the past decade, the need for such tools to aid in evaluation has become apparent. It is hoped that extension of this model to specific existing systems will be accomplished in the near future.

Acknowledgement

The authors wish to acknowledge the helpful comment and criticism of Dr. Bruce Wald, U.S. Naval Research Laboratory. The authors also wish to acknowledge the help of Mr. Albert Sacca of Analytics Incorporated in the programming and exercise of the model.

References

1. Maron, M.E., and Kuhns, J.L., "On Relevance, Probabilistic Indexing and Information Retrieval", Journal of the ACM, Volume 7, 1960, pp. 216-244.

2. Stiles, H.E., "The Association Factor in Information Retrieval", Journal of the ACM, Volume 8, 1961, p. 271.

3. Shumway, R.H., "On the Expected Gain From Adjusting Matched Term Retrieval Systems". Communications of the ACM, Volume 10, 1967, No. 11, p. 722.

4. Jones and Needham, "Automatic Term Classification and Retrieval", Information Storage and Retrieval, Volume 4, 1968.

Appendix - Computation of Fraction of Search

The Monte Carlo computations for F(r), equation (10), are here briefly described.

The denominator M(S+) of (10) is $(\frac{1}{2})^n$ times the measure of the (n-1)-dimensional "surface" of the n-dimensional unit sphere or, for n=2m (only even values of n were used),

$$M(S+) = \frac{2^{-n} \quad 2\Pi^m}{(m-1)!}, \; n/2 = m \text{ integral.} \quad (11)$$

Only the numerator $\bar{M}(\theta/T)$ need be discussed.

If Q is at an angular distance greater than θ from the nearest point on the boundary of S+, then $M(\theta/Q)$ of (5) becomes

$$M(\theta/Q) = K\int_0^\theta \sin^{n-2} \phi \, d\phi \,, \quad (12)$$

where

$$K = \Pi^{m-1} 2^{n-1} \frac{(m-1)!}{(n-2)!}, \; n/2 = m \text{ integral} \,. \quad (13)$$

If Q is at an arbitrary distance from the boundary of S+,

$$M(\theta/Q) = K\int_0^\theta f(\phi/Q) \sin^{n-2} \phi \, d\phi \quad (14)$$

where $f(\phi/Q)$ is the fraction of the "ring" at angular distance ϕ from Q that lies in S+. Then the desired numerator in (10) is

$$\bar{M}(\theta/T) = K\int_0^\theta \bar{f}(\phi/T) \sin^{n-2} \phi \, d\phi \quad (15)$$

where $\bar{f}(\phi/T)$ is the mean fraction of the ring at angular distance ϕ from Q that lies in S+, averaged over the distribution T for Q.

Equivalently, $\overline{T}(\emptyset/T)$ is the mean probability, averaged over the distribution T, that a step of angular-distance size $\emptyset$ from Q in a random direction will land in S+. Also equivalently, $\overline{T}(\emptyset/T)$ is the probability that when a Q is drawn from T and a random direction (uniformly distributed) is selected along the surface of the unit n-sphere at Q, the boundary of S+ is at an angular distance of at least $\emptyset$ in that direction. Finally, if Q is drawn from T and a random direction (uniformly distributed) is selected along the surface of the n-sphere at Q and the angular distance in that direction to the boundary of S+ is measured, the cumulative distribution function of the measured angular distance is

$$P(\emptyset) = 1-\overline{T}(\emptyset/T),$$

or

$$\overline{T}(\emptyset/T) = 1-P(\emptyset), \tag{16}$$

where $P(\emptyset)$ is the cumulative distribution function of the angular distance to the boundary as just defined. This last interpretation of $\overline{T}(\emptyset/T)$ is the one used in the Monte Carlo computations.

First the point Q_1 was drawn from a uniform distribution over the positive orthant of the k-dimensional sphere

$$x_1^2 + \ldots + x_k^2 = 1 \tag{17}$$

as follows: each of the $x_j = (x_1,\ldots,x_k)$ was taken to be the sum of six independent random numbers uniformly distributed over the interval $(-\frac{1}{2},\frac{1}{2})$. Each x_j was therefore approximately a sample from a normal distribution with zero mean. By the relevant property of normal distributions, the set $(x_1,\ldots,x_k)$, regarded as the coordinates of a k-sphere, was a sample from

a k-dimensional normal distribution with spherical symmetry. The normalization

$$q_i = \frac{|x_i|}{(x_1^2 + \ldots + x_k^2)^{\frac{1}{2}}}, \quad i=1,\ldots,k \tag{18}$$

gave the desired $Q_1 = (q_1,\ldots,q_k)$. Setting

$$q_i = 0, \qquad i=k+1,\ldots,n \tag{19}$$

gave the $Q = (q_1,\ldots,q_n)$ as defined in connection with equations (6) and (7).

Next, a direction along the surface of the n-sphere from Q might be found ("might" because a more economical way is given below) by locating the point C as follows: draw a point $(x_1,\ldots,x_n)$ from an n-dimensional normal distribution with spherical symmetry by taking each x_j to be the sum of six independent random numbers evenly distributed over $(-\frac{1}{2},\frac{1}{2})$. Compute

$$L = q_1x_1+\ldots+q_nx_n, \tag{20}$$

and set

$$c_i = \frac{x_i}{L}, \qquad i=1,\ldots,n. \tag{21}$$

QC is then tangent to the unit sphere and OQC is a right angle, because, as may be verified,

$$\sum_{i=1}^{n} q_i(c_i - q_i) = 0. \tag{22}$$

However, if the procedure of (20), (21) and (22) is followed just as outlined, the angular distance from Q in the direction of QC will be zero with probability $1-2^{-n+k}$, because that

distance is zero if any of $x_{k+1},\ldots,x_n$ are negative. To save computation, therefore, $x_{k+1},\ldots,x_n$ were constrained to be positive by a method equivalent to using

$$c_i = \frac{|x_i|}{L}, \qquad i= k+1,\ldots,n \qquad (23)$$

in place of part of (21). The compensation

$$P(\emptyset) = 1-2^{-n+k}(1-p^*(\emptyset)) \qquad (24)$$

was applied for the distortion in the sampling represented by (23), where $p^*(\emptyset)$ was the observed cumulative distribution function found using (23).

Using (16) and (24) in place of (14) gives:

$$\overline{M}(\theta/T) = K\int_0^\theta (1-p^*(\emptyset)) \sin^{n-2} \emptyset \, d\emptyset. \qquad (25)$$

Returning to the problem of determining $p^*(\emptyset)$ for cumulative distribution function of the angular distance $\emptyset$ from Q to the boundary of S+ in the direction QC, consider the point $X(t) = (x_1,\ldots,x_n)$ where

$$x_i = (1-t)q_i+tc_i, \quad i=1,\ldots,n. \qquad (26)$$

For t=0, X(t)=Q and for t=1, X(t)=C. The point X(t) moves continuously from Q to C in the direction QC as t increases. For large enough t, except in unlikely special cases that need not be discussed here nor be guarded against in the computations, at least one of the coordinates will pass through zero. The smallest positive value of t for which this occurs marks the point X on the line QC where that line leaves the positive orthant of the space. This smallest positive value is found by solving the n equations

$$0 = (1-t)q_i + tc_i, \quad i=1,\ldots,n \tag{27}$$

for t and noting the smallest non-negative solution. Using this value of t in (26) gives the coordinates of X. The resulting value of $\emptyset$ is

$$\emptyset = \arccos \frac{1}{(x_1^2 + \ldots + x_n^2)^{\frac{1}{2}}}, \tag{28}$$

because OQX is a right angle and

$$\cos\emptyset = \frac{\overline{OQ}}{\overline{OX}}. \tag{29}$$

One thousand independent sample values of $\emptyset$ as in (28) were computed for each curve of F(r) versus r in the plotted results. The values of $\emptyset$ were ordered and numbered such that

$$\emptyset_1 \leq \emptyset_2 \leq \cdots \leq \emptyset_{1000}. \tag{30}$$

The quadrature of (25) was approximated by a sum in the obvious way using 1000 terms, the j^{th} representing the interval $\emptyset_{j-1} \leq \emptyset < \emptyset_j$, with $p^*(\emptyset)$ taken equal to $(j-1)/1000$ over the interval. Together with (11), (25) gives the result (10) for F(r), the fraction of the subfile that must be searched.

AN EXPERIMENT TO MEASURE
DEPTH OF FILE SEARCH
IN CLUSTERED FILES

by: Andrew Noetzel

AN EXPERIMENT TO MEASURE
DEPTH OF FILE SEARCH IN
CLUSTERED FILES

This document describes a generalized model of an information storage and retrieval system and an algorithm for determining the depth to which a file must be searched to overcome indexing and querying noise. To make the results of the study relevant to real information retrieval systems, the model will be provided with statistical parameters taken from actual files.

The general model proposed here is a combination of two specific models that have been used to describe information retrieval systems, the Boolean model and the probabilistic model. These two models will be presented in order to demonstrate the general model.

The Boolean Model

In the Boolean model, a file in an information retrieval system is represented by an m-dimensional space, in which each dimension represents an index term used in the file. The total number of index terms used in the file is therefore m. Let the sequence $t_1, t_2, \ldots, t_m$ be an ordering of the terms of the file.

A document D_i, stored in the file, is represented by the vector $V_i=(a_{i1},a_{i2},\ldots,a_{in})$ where each element a_{ij} will have the value one if D_i is indexed by term t_j, and zero if not.

Since a document may be indexed by any subset of the m terms of the file, the documents are potentially distributed throughout all of the 2^m points which represent the corners of the m-dimensional unit hypercube.

For consistency with previous work, all the documents are assumed to be equally worthy of retrieval given only that the correct question is asked; therefore, each document vector is normalized to length one.

<u>The Probabilistic Model</u>

In the probabilistic model, the file is described by the same m-dimensional hyperspace. A document D_i is represented by the same vector V_i, except that each a_{ij} now represents the probability that D_i will satisfy a request containing the term t_j, where $0 \le a_{ij} \le 1$ for all $1 \le j \le n$. It is assumed that these probabilities are completely known. Assume, for example, that the probabilities are generated by finding each term t_j that occurs in the text of D_i and assigning it some relevance number $a_{ij} (\le 1)$. Then, for each of the other terms t_k which do not appear in D_i, the number $a_{ik} \le 1$ is calculated both from co-occurrence data taken from a very large sample of text and from previously assigned values of a_{ij}. Under those conditions, the term a_{ik} will take on the value zero with the same probability that it takes on any other real value in the open interval (0,1).

The documents will then be distributed in the m-dimensional hyperspace with no finite density of documents occurring in any proper subspace of this space. As before, the length of each document vector V_i may be normalized so that the document values are considered to be distributed on the surface of the positive orthant of the m-dimensional

hypersphere (positive orthant since the a_{ij}'s are non-negative). They will not be evenly distributed on this surface but will instead cluster close to the border of each of the subspaces. This results from many terms t_j having very small relevance to particular documents D_i and corresponding a_{ij}'s being close to zero.

The Combined Model

Merging the two models, one obtains the general model for probabilistically-indexed files. In the combined model the documents are similarly represented by the normalized vector V_i whose terms a_{ij} have values $0 \le a_{ij} \le 1$. Based upon this assignment, there is a finite distribution of documents in each subspace of the m-space. The documents are distributed on the surface of the n-dimensional hypersphere of each n-dimensional subspace of the m-space.

The algorithm developed measures the amount of the file that falls within the solid angle b of a query vector. In making this measurement, the effects of clustering upon the distribution of documents in the m-space that are known to occur when documents are assigned index terms must be taken into account.

In the general model, the clustering effects are represented by the varying density of documents populating subspaces. For the moment, assume that the distribution of documents in each subspace is known from data taken from an actual file. In order to save computation, we would like to analyze only a part of the file to obtain results applicable to the entire file. We might begin by selecting one cluster (that is, a relatively heavily-populated subspace) and work within it, ignoring the remainder of the file. This approach

does not reflect the clustering properties, because no matter what subspace is chosen there will be a significant population of documents which cannot be completely described by the n index terms of the cluster and yet are relevant to it. As an example of this problem consider a document which has relevant all n terms of the cluster plus one more.

The results of such an analysis determine the depth to which the cluster must be searched. This depth is not a useful result unless the size of the cluster relative to the entire file can be measured, and unless it is assured that the document sought will be in the cluster. In general, this will not be true. Since, then, the environment of a cluster must be considered in order to determine the effects of the cluster, the isolation of an n-dimensional subspace representation of the cluster in the development of the model is not sufficient.

It will be more useful to base the calculation on the set of index terms that result from a query. This set may be equivalent to some meaningful statistical cluster of index terms in the data base, it may be a subset of it, or it may be somewhat different from it.

The Approach to the Analysis Problem

The approach taken is to first generate the query vector and to concern ourselves with the distribution of documents in the file immediately surrounding the query vector. The distribution of document vectors throughout the file is generated next, and the document density in the subspaces surrounding the query vector is recorded. The document vectors will be random variables whose distributions are compiled from statistics taken from a real file. No attempt will be made to isolate or identify the clusters, but the

distribution from which the documents are generated will ensure that clustering effects are present.

The procedure is composed of three steps. First, some statistics showing the actual distribution of index terms and their interrelations must be obtained for input to the model. Secondly, after the query vector has been generated, the fraction of the file specified by that query vector must be determined. How this fraction of the file is distributed throughout the subspaces of the space identified by the query vector must also be found. Finally, beginning with the query vector, the fraction of the subfile encompassed within the solid angle b of the query vector will be determined as a function of b. This last step of the work is similar to the existing model (ref. previous paper) except that the search will not be limited to one surface in n-space. Instead the search will be extended to each of the n surfaces of dimension n-1, then to each of the $n \cdot (n-1)/2$ surfaces of dimension n-2, and so on, until it encompasses the entire set of all subspaces of the query vector space.

The three steps are described in detail in the following paragraphs.

<u>Step I</u>. We will first consider the size of the subfile that is implicated by a randomly-chosen query vector, that is, what portion of the total file would be obtained if every document were retrieved whose set of index terms had any term in common with the terms in the query vector's set. For small files, the portion retrieved is dependent upon the size of the file: as the size of the file increases, the total number of index terms increases proportionally, and the portion of the file represented by a single term

decreases. On the other hand, for larger files the total number of index terms tends to remain constant as the file size varies. Therefore the statistics used in this model will be taken from a reasonably large file, and since the results will be expressed as fractions of total file, they will be applicable to all large files. It is assumed that the document distributions, normalized by file size, remain the same for all large files.

A query vector is composed of n randomly-selected terms. To determine the size of the subfile implicated by these n terms the model must contain some indication of the number of documents in the file that have been indexed by exactly this combination of n terms, and the number which have been exactly indexed by each of 2^n subsets of this set of n terms. This implies that the distribution of documents throughout every combination of the total number of index terms in the file, m, must be known. For moderate size files, m will range from 200 to 1000 terms. The model must then contain 2^{200} to 2^{1000} data items, thus making files of this size far too large to be considered.

An approximation to the total amount of information contained in a full description of the document distributions throughout the total file's m-space can be derived from the record of the distribution of the documents over the index terms taken one at a time and two at a time. These statistics are available from many reports on information retrieval systems. It is assumed, then, that the file is described by a list of m terms giving the absolute probabilities of the appearance of an index term t_i in a document,

$$P(t_i) = \frac{\text{Number of documents indexed by } t_i}{\text{Total number of documents}}$$

and by another list of $m(m-1)/2$ terms, giving the probability of co-occurrence of all pairs of index terms, t_i and t_j in a document,

$$P(t_i t_j) = \frac{\text{Number of documents indexed by } t_i \text{ and } t_j}{\text{Total number of documents}} .$$

Since these fractions are indications of the frequency of use of the basis vectors of the m-space, individually and in pairs, they indicate the directions taken by document vectors in m-space. The indications of the length of the document vectors is given by the distribution of the number of index terms per document in the file. This discrete probability distribution, called $N(x)$, is obtained by sampling the number of index terms assigned to documents in a real file.

<u>Step II</u>. After a random query vector has been generated, the density of documents falling into the n-dimensional subspace identified by the n-term query vector is calculated. This density must include documents whose total description vector lies outside the query subspace, but which have some terms in common with the query. Each such document is projected into the k-dimensional subspace of the query space where k is the number of terms the document and query have in common.

The algorithm for generating document vectors is as follows:

(1) A random number x is generated, and the distribution $N(x)$ is used to determine the number of terms T by which the document is defined.

(2) The list of probabilities $P(t_i)$ $(1 \leq i \leq m)$ is

used as a second distribution to select a particular index term. Assume term t_{a1} is chosen.

(3) If $T>1$ a second index term is needed to describe the document. The conditional probabilities of selecting term t_j given that t_i has been selected can be calculated from the list of co-occurrence probabilities:

$$P(t_i/t_j)=P(t_i t_j)/P(t_i).$$

Therefore, the list of probabilities $P(t_i/t_{a1})$ is used as a distribution to select a second index term. Assume term t_{a2} is selected as the second term.

(4) If $T>2$, a third index term is needed. It should be selected from the conditional probability $P(t_k/t_i t_j)$; however, this is not available. It may be approximated by the geometric mean of the conditional probabilities $P(t_i/t_a)$ and $P(t_i/t_b)$ that are available from the co-occurrence probabilities. The list

$$P(t_i/t_a t_b)=\sqrt{P(t_i/t_a)\cdot P(t_i/t_b)}$$

is used as a distribution to select the third term.

(5) Each additional index term, up to T, is selected by a probability distribution derived from the conditional probabilities given the previously selected terms. The above approximation is generalized. Thus, for the $k+1^{st}$ term:

$$P(t_i/t_{a1}t_{a2},\ldots,t_{ak}) =$$

$$\sqrt[k]{P(t_i/t_{a1})\cdot P(t_i/t_{a2}),\ldots,P(t_i/t_{ak})}.$$

For a particular query vector, the subspaces of interest in the index-term-space are identified as follows. The n terms of the query vector are ordered; if a particular subspace has the dimension corresponding to a particular term, the binary form of its identifying number will have a one in the position of that index term following the ordering of the query vector terms, otherwise it will be zero. Each subspace as used here does not include any of its subspaces, thus the set of subspaces partitions the set of documents.

Each document vector generated according to this procedure will either fall into one of the 2^n-1 subspaces of the query vector space (not including the null space) or it will be projected into one of the 2^n subspaces of the query vector space. Therefore, 2^n counters, C_0 to C_{2^n-1} will be maintained and will count the number of documents which either fall onto or are projected onto the surface of each of the corresponding subspaces S_0 to S_{2^n-1}. The counter corresponding to the zero-dimension subspace will count all documents not implicated by the query.

Let D be the total number of documents generated. C_0 of the D documents have no index term in common with the query, that is, they fall into the 0-dimension subspace. The query then implicates a fraction $(D-C_0)/D$ of the total file.

Each subspace S_i of the file will contain a density of documents $D_i=C_i/D$ relative to the entire file.

Step III. In the last step of the procedure, the portion of the document space that either falls onto or is projected onto the surface subtended by the solid angle b is calculated as a function of the angle b.

Let b_i be the angular distance from the query vector Q to the border of the i^{th} n-1 space, which is the subspace identified by the subspace number 2^n-1-2^{n-i}. Let

$$b_{min}=\min\left\{b_1 b_2 \ldots b_n\right\}.$$

For $b<b_{min}$, the part of the file encompassed within angle b of Q is proportional to the surface area in the first orthant in n-space subtended by b. We call this quantity S(b,n). The total surface area in the first orthant in n-space is S(n). Thus, the measure of the file encompassed within angle b is

$$F(b)=\frac{S(b,n)}{S(n)}\cdot D_{2^n-1},$$

where D_{2^n-1} is the density of documents on the n-space surface.

Now, we will increase the solid angle b beyond the border of the closest subspace. If $b_k=b_{min}$, the closest subspace will be the subspace with the k^{th} dimension missing. Its identifying number will be

$$a=2^n-1-2^{n-k}$$

found by subtracting the one in the k^{th} bit position of the binary-form identifying number.

Let Q_a be the projection of Q into S_a. Let b_a

be the angle from Q_a which subtends the same n-1 dimensional surface as does the angle b. Then b and b_a are related by:

$$\cos^2 b_k + \cos^2 b_a = \cos^2 b$$

Thus

$$b_a = \cos^{-1}\sqrt{\cos^2 b - \cos^2 b_k} \quad .$$

When b is increased beyond the border of the closest subspace, but not as far as the next-closest border, the depth of the file encompassed by b also includes the contribution of the file in n-1 space that is subtended by the angle b_a. This contribution is

$$\frac{S(b_a, n-1) \cdot D_a}{S(n-1)}$$

Thus

$$F(b) = \frac{S(b,n)}{S(n)} \cdot D_{2^n-1} + \frac{S(b_a, n-1)}{S(n-1)} \cdot D_a \quad .$$

As the angle b is expanded still further, it will either reach the border of another n-1 space, or else the angle b_a will reach the border of an n-2 space.

For the first case, we identify

$$b_j = \min(b_1, b_2, \ldots, b_{k-1}, b_{k+1}, \ldots, b_n)$$

and the next subspace is

$$c = 2^n - 1 - 2^{n-j} \quad .$$

For the second case, we measure the angular distance of Q_a to each of the (n-1) n-2 space borders. Let $b_{a1}, b_{a2}, \ldots, b_{an}$ be these angular distances, including an arbitrarily large value for b_{ak} (which now has no meaning),

in order to keep the same ordering.

Suppose $b_{ap}=\min\{b_{a1}, b_{a2}, \ldots, b_{an}\}$. Then the first n-2 space encountered is identified by $d=2^{n-1}-1-2^k-2^p=a-2^p$.

As angle b increases, it must be determined whether

$$b > b_j \text{ or } b_a > b_{ap}$$

will occur first.

Using the definition of b_a, this implies that if

$$b_j < \cos^{-1}\sqrt{\cos^2 b_{ap} - \cos^2 b_k},$$

this first case will occur first.

Determining the depth of file for the first case, let Q_c be the projection of Q into space S_c, and

$$b_c = \cos^{-1}\sqrt{\cos^2 b - \cos^2 b_j} \quad .$$

The file encompassed by the angle b will include the contribution in this sub pace; thus

$$F(b) = \frac{S(b,n)}{S(n)} \cdot D_{2^n-1} + \frac{S(b_a,n-1)}{S(n-1)} \cdot D_a + \frac{S(b_c,n-1)}{S(n-1)} \cdot D_c \quad .$$

Returning to the second case, let Q_d be the projection of Q_a into the n-2 space d, and b_{ad} be the angle in space d subtending the same surface area as the angle b_a:

$$b_{ad} = \cos^{-1}\sqrt{\cos^2 b_a - \cos^2 b_{ap}} \quad .$$

Then, the contribution to the file encompassed by the angle b in this space is

$$\frac{S(b_{ad},n-2)}{S(n-2)} \cdot D_d ,$$

and the total file encompassed by the angle b is

$$F(b) = \frac{S(b,n)}{S(n)} \cdot D_{2^n-1} + \frac{S(b_a,n-1)}{S(n-1)} \cdot D_a + \frac{S(b_{ad},n-2)}{} \cdot D_d \ .$$

Ultimately, as angle b is increased sufficiently, the last term from both of the above expressions for F(b) will become included in the equation for F(b):

$$F(b) = \frac{S(b,n)}{S(n)} \cdot D_{2^n-1} + \frac{S(b_a,n-1)}{S(n-1)} \cdot D_a$$

$$+ \frac{S(b_{ad},n-2)}{S(n-2)} \cdot D_d + \frac{S(b_c,n-1)}{S(n-1)} \cdot D_c.$$

It is obvious that as the angle b is increased further, more subspaces will be encompassed within angle b, and F(b) will have even more terms. A general structure is required which will include the present and future contributions to the file for each subspace. The possibility of programming the computation of F(b) in a recursive program is obvious, since the computation of the contribution of each subspace of dimension k is the same as that of the subspace of dimension k+1.

Finally, it is expected that a significant result for the F(b) relation will be obtained long before all the subspaces of the n-space are included. Therefore, an approximation to this procedure will be obtained if only the cases of subspaces of degree n-1 and n-2 are considered. In this case, it is not necessary for the program which performs the calculations to be recursive. All combinations that may be required for calculations within subspaces may be represented explicitly.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Analytics, Incorporated 179 Washington Lane Jenkintown, Pa. 19046	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP --NA--

3. REPORT TITLE

QUANTIFICATION OF INFORMATION STORAGE AND RETRIEVAL METHODOLOGIES

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Interim Report 15 September 1969 - 14 May 1970

5. AUTHOR(S) (Last name, first name, initial)

Plotkin, Morris   Noetzel, Andrew
Epstein, Samuel D.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
5 June 1970	35	4

8a. CONTRACT OR GRANT NO. ONR N00014-70-C-0044	9a. ORIGINATOR'S REPORT NUMBER(S) 1013.1-1
b. PROJECT NO. NRL RF018-02-41 c. NRL RR003-09-41-502 d.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) ---None---

10. AVAILABILITY/LIMITATION NOTICES

Qualified requesters may obtain copies of this report from DDC.

11. SUPPLEMENTARY NOTES --None--	12. SPONSORING MILITARY ACTIVITY Office of Naval Research Washington, D.C. NRL Requisition

13. ABSTRACT

This Interim Report discusses the development of a tool for evaluating the efficiency of probabilistic information retrieval systems as a function of human variability noise. Based upon the variation of people in indexing documents and specifying information requests, this model uses Monte Carlo computations to compare file search depth to total noise. The results presented are for an idealized system, although the technique is applicable to actual cases. A method whereby statistics of true systems may be used to extend the model is discussed. The tool developed can be of aid in selecting desired systems for a parti ular application.

DD FORM 1473

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Information Storage and Retrieval						
Retrieval Methodologies						
File Search						
Indexing						
Probabilistic IS&R						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.